



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/090,341

03/04/2002

Cary Lee Bates

ROC920010348US1

3890

46797

7590

10/20/2006

IBM CORPORATION, INTELLECTUAL PROPERTY LAW  
DEPT 917, BLDG. 006-1  
3605 HIGHWAY 52 NORTH  
ROCHESTER, MN 55901-7829

EXAMINER

FRANCIS, MARK P

ART UNIT

PAPER NUMBER

2193

DATE MAILED: 10/20/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

**Office Action Summary**

Application No.

10/090,341

Applicant(s)

BATES ET AL.

Examiner

Mark P. Francis

Art Unit

2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 04 August 2006.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-20 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-20 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)                                | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                       | 5) <input type="checkbox"/> Notice of Informal Patent Application                       |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

### DETAILED ACTION

1. This action is responsive to the amendment filed August 04, 2006.
2. Per applicants' request, claims 1-20 remain pending.

### ***Claim Rejections - 35 USC § 103***

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claims 1-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wimble(U.S. Pat. 5,812,850) in view of Warmink.(U.S. Pat. 6,611,924)

Regarding claims 1 and 9,

Wimble discloses a method of debugging executable code configured to access associated data in a data repository(Col 6:33-47, "...Debugging information is really a database of information....", Col 8:16-27, "...the Debugger 48, in the same database..."), comprising:

Initiating a debugging session for the executable code, and during the debugging session;(See Abstract, lines 108, "provides an interactive...")(Col 3:35-38, "the debugging session...")

Art Unit: 2193

Monitoring step-by-step execution of the executable code;(Col 9-11:1-67, "...The Instruction Map 62 provides detailed knowledge of how each executable machine instruction...")

determining whether the monitored executable code has accessed the associated data in the data repository(Col 12:47-67, "...a component in the Debugger Database...") but does not disclose if so, determining whether to display the associated data on the basis of whether the associated data is restricted data; wherein determining whether to display the associated data comprises referencing predefined access restriction rules defining at least one rule preventing at least a portion of the associated data from being displayed to unauthorized users; and

upon determining not to display the associated data on the basis of the referenced predefined access restriction rules, outputting masking characters on an output screen indicative of the associated data without revealing a value of the associated data, whereby selected data from the data repository is concealed from a user debugging the executable code.

Warmink discloses determining whether to display the associated data on the basis of whether the associated data is restricted data; wherein determining whether to display the associated data comprises referencing predefined access restriction rules defining at least one rule preventing at least a portion of the associated data from being displayed to unauthorized users(Col 2:35-67, "...provides a filterable debug output...", Col 3:30-40, "...to further prevent unauthorized users...")

Art Unit: 2193

and

upon determining not to display the associated data on the basis of the referenced predefined access restriction rules, outputting masking(Col 2:35-67, "...developer-selected masks...") characters on an output screen indicative of the associated data without revealing a value of the associated data, whereby selected data from the data repository is concealed from a user debugging the executable code(Col 2:35-67, "...based on some specified rules of filtering...") in an analogous system for the purpose of providing debug output strings, each debug output statement comprising a fixed-size text string that is replaced with a unique text string number to provide stripped source code.(Warmink:Col 3:43-51)

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to include a function for determining whether to display data based upon predefined access restriction rules to Wimble's invention.

The modification would have been obvious because one of ordinary skill in the art would have been motivated to provide debug output strings, each debug output statement comprising a fixed-size text string that is replaced with a unique text string number to provide stripped source code.(Warmink:Col 3:43-51)

Regarding claim 16,

Art Unit: 2193

Wimble discloses a computer-readable medium containing a debug program which, when executed, performs an operation of debugging code configured to access associated data in a repository(See Above rejection of claims 1 and 16), the debug program comprising:

A debugger user interface; (e.g. See Fig. 1, element 22 and Col 6:48-65, "...an interface between Information Providers and Information Consumers...")

A debug engine configured to selectively pass data to the debugger user interface(Col 26:21-25, "...a debugger engine..." and See Fig. 3, element 54 and related text) but does not disclose according to predefined access restriction rules defining at least one rule prohibiting at least a portion of the associated data from being displayed to a user operating the debug program, whereby selected data from the data repository is concealed from the user debugging the executable code.

Warmink discloses to selectively pass data to the debugger user interface according to predefined access restriction rules(Col 2:35-67, "...developer-selected masks...") defining at least one rule prohibiting at least a portion of the associated data from being displayed to a user operating the debug program, (Col 2:35-67, "...based on some specified rules of filtering...") whereby selected data from the data repository is concealed from the user debugging the executable code in an analogous system for the purpose of providing debug output strings, each debug output statement comprising a

Art Unit: 2193

fixed-size text string that is replaced with a unique text string number to provide stripped source code.(Warmink:Col 3:43-51)

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to include a function for determining whether to display data based upon predefined access restriction rules to Wimble's invention.

The modification would have been obvious because one of ordinary skill in the art would have been motivated to providing debug output strings, each debug output statement comprising a fixed-size text string that is replaced with a unique text string number to provide stripped source code.(Warmink:Col 3:43-51)

#### Dependent claims

Regarding claims 2 and 10,

Wimble discloses wherein determining whether to display the associated data comprises determining whether the associated data can be provided to a debugger user interface.(Col 6:33-47, "...It is not often that the information...", Col 7: 12-38, "...the information provider's interface to the symbolic information...", See Fig. 3 elements 60(interface), 61(data), 63(data))

Regarding claims 4 and 12,

wherein determining whether the associated data can be displayed comprises referencing a restricted data table(Wimble: Col 28:3-5, "said information database") created in response to reading the associated data from the repository and according to the predefined access restriction rules. (Warmink: Col 2:35-67, "...provides a filterable debug output...", Col 3:30-40, "...to further prevent unauthorized users...")

Regarding claim 5,

Wimble discloses wherein determining whether to display the associated data is performed by a debugging program.(Col 8:16-36, "...the Debugger to control the debugged program...")

Regarding claim 6,

determining whether to display the associated data is performed by a debugging program.(Wimble: Col 8:16-36, "...the Debugger to control the debugged program...") implementing the predefined access restriction rules. (Warmink:Col 2:35-67, "...provides a filterable debug output...", Col 3:30-40, "...to further prevent unauthorized users...")

Regarding claims 11 and 17,

The debug engine is configured to:

Determine that the associated data cannot be displayed during the debugging session;(Wimble: Col 19: 11-15, "...the debugger couldn't determine...") and conceal



Art Unit: 2193

the display of the associated data by displaying text characters on an output screen indicative of the associated data without revealing a value of the associated data.

(Warmink:Col 6:50-67, "...source code is replaced with a unique number, such as ENUM...")

Regarding claim 15,

The executable code accesses the associated data comprising a record and wherein determining whether the associated data can be displayed comprises:

Referencing the predefined access restriction rules defining at least one rule preventing at least one field value from being displayed;(Warmink:Col 6:50-67, "...source code is replaced with a unique number, such as ENUM...")

and

Determining whether the record contains the at least one field value.(Warmink:Col 6:50-67, "...source code is replaced with a unique number, such as ENUM...")

Regarding claim 19,

The at least one rule defines a value and an associated value, wherein if the associated value has been displayed the debug engine will not provide the value to the debugger user interface for display. (Warmink:Col 6:50-67, "...source code is replaced with a unique number, such as ENUM...")

Regarding claim 18,

Art Unit: 2193

Wherein the debug engine is configured to selectively pass data to the debugger user interface by referencing a restricted data table created in response to reading the associated data from the repository and according to the predefined access restriction rules.(See above rejection of claims 4 and 12)

5. Claims 8, 14, and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wimble in view of Warmink and further in view of Kolawa.(U.S. Pat 6,085,029)

Regarding claims 8 and 14.

Neither Wimble nor Warmink disclose determining whether to display the associated data comprises referencing a parse expression defining a data format and an output expression defining a restricted portion of the parse expression.

Kolawa discloses determining whether to display the associated data comprises referencing a parse expression(Col 5:46-50, "a parsing...") defining a data format and an output expression defining a restricted portion of the parse expression(Col 7:28-35, "...form of debug output...") in an analogous system for the purpose to generate code for the target program which not only functions as was originally intended, but also contains calls to instrumentation procedures which provide automatic error detection of dynamic program errors as well as an ability to automatically generate test cases.

(Kolawa: Col 7: 36-41)

Art Unit: 2193

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to reference a parse expression for determining data to be displayed during a debugging session.

The modification would have been obvious because one of ordinary skill in the art would have been motivated to develop a method of automatically instrumenting a computer program for dynamic debugging as an integral part of the program development cycle and without introducing an extra stage in the program development cycle. (Col 2: 27-33, "...an extra stage...")

Regarding claim 20

Neither Wimble nor Warmink disclose that the at least one rule defines a parse expression defining a data format and an output expression defining a restricted portion of the parse expression, whereby all values having restricted portion will not be provided to the debugger user interface for display.

Kolawa discloses that the at least one rule defines a parse expression defining a data format and an output expression (Col 7:28-35, "...form of debug output...") defining a restricted portion of the parse expression (Col 15:49, "parse tree..."), whereby all values having restricted portion will not be provided to the debugger user interface for display (Col 16:53-61, "...interface is inserted...") in an analogous debugging program for the purpose to generate code for the target program which not only functions as was

Art Unit: 2193

originally intended, but also contains calls to instrumentation procedures which provide automatic error detection of dynamic program errors as well as an ability to automatically generate test cases. (Kolawa: Col 7: 36-41)

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to reference a parse expression, whereby all values that contain restricted portion will not be provided for determining data to be displayed during a debugging session.

The modification would have been obvious because one of ordinary skill in the art would have been motivated to develop a method of automatically instrumenting a computer program for dynamic debugging as an integral part of the program development cycle and without introducing an extra stage in the program development cycle. (Col 2: 27-33, "...an extra stage...")

### ***Response to Arguments***

6. Applicant's arguments filed on April 04, 2006 have been fully considered but they are not persuasive. Following is the Examiner's response to Applicants' arguments.

With respect to claims 1 and 9, Applicant essentially argues that Wimble et al. does not anticipate the features of this claim because Wimble et al. does not teach or suggest the executable code accessing the debugger database.

In response, the Examiner disagrees Note Col 8, lines 16-49, it is here that Wimble et. al discloses symbolic properties of the program components that contain data that change during program execution. Wimble also discloses that the symbolic properties change themselves on demand to generate the symbolic data that is stored within. In addition Wimble discloses, a database that includes a collection of Symbolic Elements which model a program that is to be built, and maintain a description of the program when it is built. Thus, Wimble does disclose the executable code accessing the debugger database.

With respect to claims 1 and 9, Applicant argues that Warmink does not teach that the limitation of determining whether the monitored executable code has access the associated data in the data repository, and if so, determining whether to display the associated data on the basis of whether the associated data is restricted data.

In response, The Examiner disagrees, Note Col 2 25-60, it is here that Warmink discloses that each trace statement provides a filterable debug output, that can be selectively enabled based on certain masks. Further Warmink teaches that each program under development may have several major software or code objects or portions which may have a different code developer. Each such object may be

Art Unit: 2193

assigned its own trace identifier so that only trace statements for a given software object are enabled. Therefore, Warmink does teach determining whether the monitored executable code has access the associated data in the data repository, and if so, determining whether to display the associated data on the basis of whether the associated data is restricted data.

With respect to claim 1, Applicant essentially argues that Warmink does not teach or suggest outputting masking characters on an output screen indicative of the associated data without revealing a value of the associated data.

In reply, the Examiner differs, Note Col 635-60, it is here that Warmink teaches that each of the fixed text field inside each of the debug output statement is replaced with a unique number such as an ENUM, which is the masking character in this instance. Thus, Warmink does teach outputting masking characters on an output screen indicative of the associated data without revealing a value of the associated data.

Regarding claim 16, Applicant essentially argues that Wimble in view of Warmink does not teach or disclose a debug engine configured to selectively pass data to the debugger user interface according to predefined access restriction rules defining at least one rule prohibiting at least a portion of the associated data from being displayed to a user operating the debug program, whereby selected data from the data repository is concealed from the user debugging the executable code.

In reply, the Examiner disagrees, Note Col 2:25-50, it is here that Warmink teaches that each trace statement provides a filterable debug output which can be selectively enabled based on certain masks. Like a print statement, a trace statement can be used to cause various messages or other debug information to be printed or displayed on a monitor. Also, Warmink discloses that each trace statement provides a filterable debug output, that can be selectively enabled based on certain masks. Further Warmink teaches that each program under development may have several major software or code objects or portions which may have a different code developer. Each such object may be assigned its own trace identifier so that only trace statements for a given software object are enabled. Therefore, Warmink does not teach or disclose a debug engine configured to selectively pass data to the debugger user interface according to predefined access restriction rules defining at least one rule prohibiting at least a portion of the associated data from being displayed to a user operating the debug program, whereby selected data from the data repository is concealed from the user debugging the executable code.

### ***Conclusion***

**5. THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within


Art Unit: 2193

TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mark P. Francis whose telephone number is (571)272-7956. The examiner can normally be reached on Mon-Fri 8:00-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571)272-3719. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.



KAKALI CHAKI  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 210



Application/Control Number: 10/090,341

Art Unit: 2193

Page 16